

xFrame5 아키텍처 가이드

2022.04.13

(주)소프트베이스

Contents

1. xFrame5 실행 환경 아키텍처
2. xAdmin5 배포 환경 아키텍처
3. xFrame5 컴포넌트 아키텍처
4. xFrame5 이벤트 아키텍처
5. xFrame5 데이터셋 아키텍처

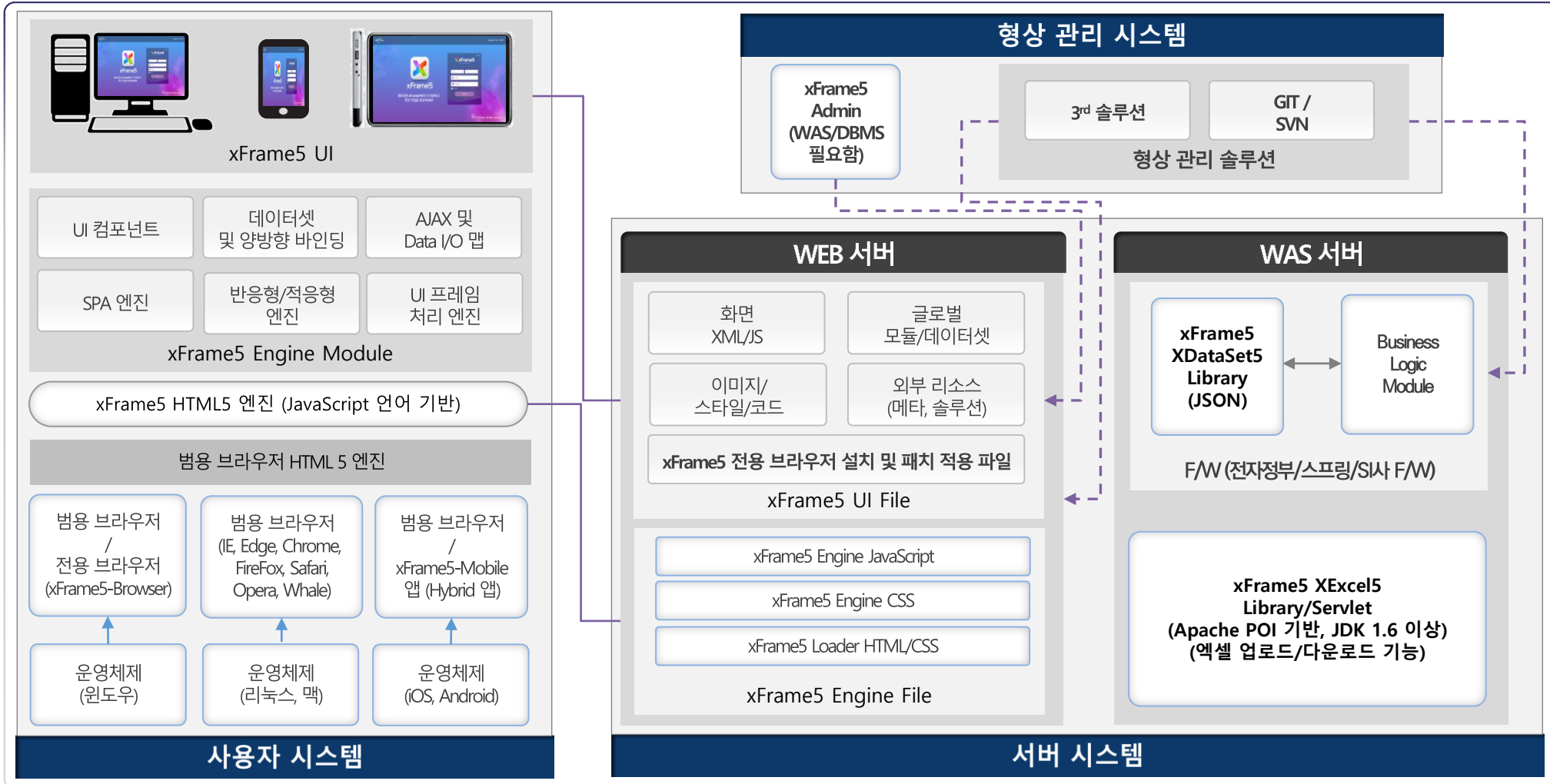


1. xFrame5 실행 환경 아키텍처

1.1. 실행 환경 아키텍처

▶ 아키텍처 상세 내용

xFrame5 솔루션의 실행 환경 아키텍처는 아래와 같이 사용자 시스템, 서버 시스템, 형상 관리 시스템으로 구성되며, 서버 시스템 내부에는 WEB/WAS 서버 S/W 내에 xFrame5 관련 파일이 구성된다.



1. xFrame5 실행 환경 아키텍처

1.1. 실행 환경 아키텍처

▶ 아키텍처 상세 내용 - 개발 아키텍처

xFrame5는 HTML5기반의 UI개발 솔루션으로 Any Device, Any Brower, Any Server Platform을 지원하는 One Source Multi Use를 지원합니다.



OSMU 개발툴 - xFrame5

- ✓ Integrated UI Development S/W
- ✓ UI 화면 설계 기능(설계서)
- ✓ UI 화면 개발 기능 (SPA 아키텍처)
- ✓ 크로스 브라우징 지원
- ✓ 반응형/적응형 디자인



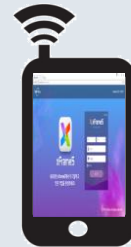
개발

XML & JavaScript



PC용 UI
(전용 브라우저)

PC용 UI
(범용 브라우저)



모바일용 UI
(Web/Hybrid APP)
(iOS, Android)



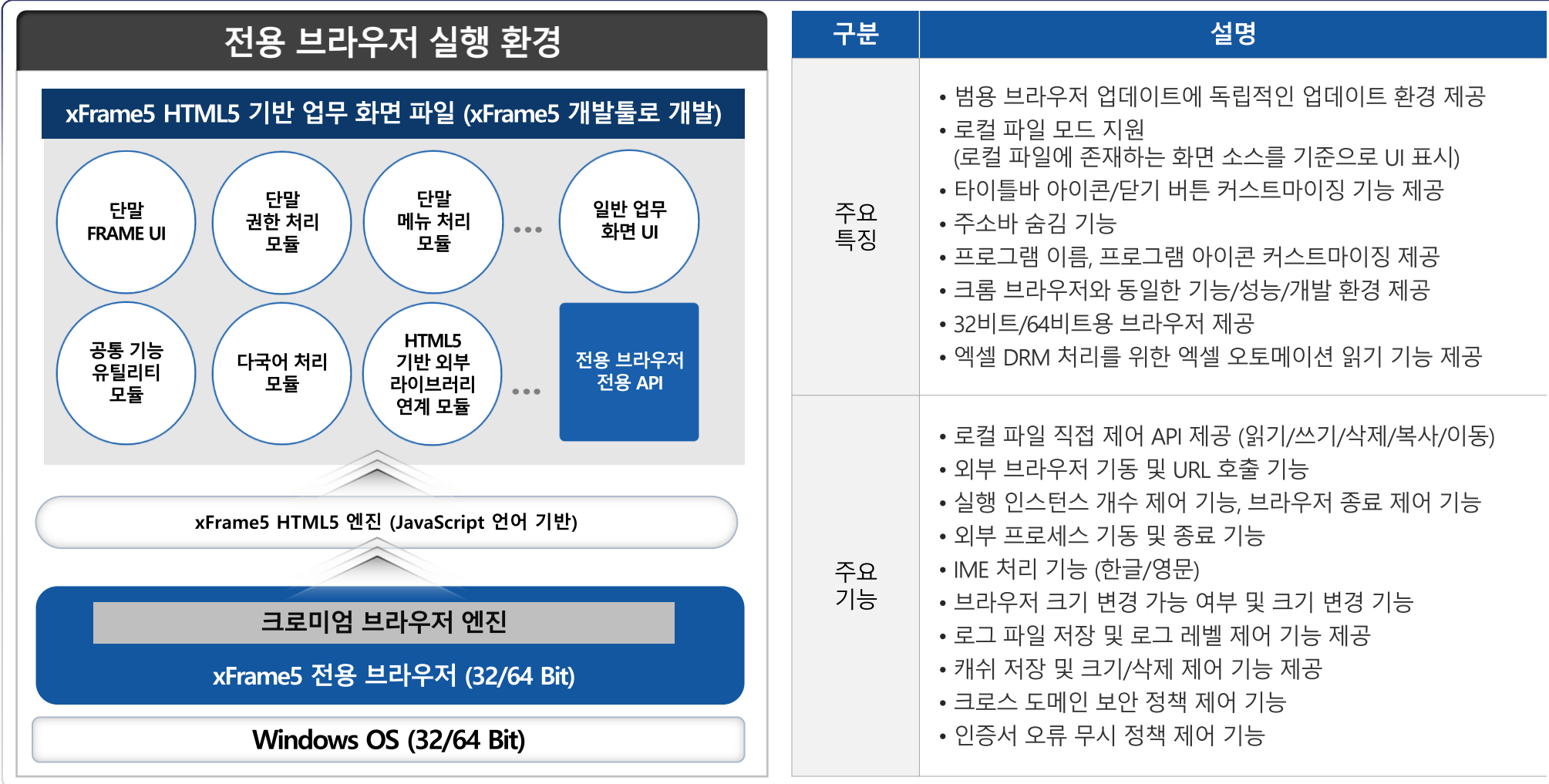
태블릿용 UI
(Web/Hybrid APP)
(iOS, Android)

1. xFrame5 실행 환경 아키텍처

1.1. 실행 환경 아키텍처

▶ 아키텍처 상세 내용 - 전용 브라우저

xFrame5 PC용 전용 브라우저는 크롬 브라우저의 기반 소스인 크로미엄 오픈 소스 기반의 엔진에 기반한 브라우저입니다. 범용 브라우저에서 제공이 불가능한 다양한 부가 기능을 제공합니다.

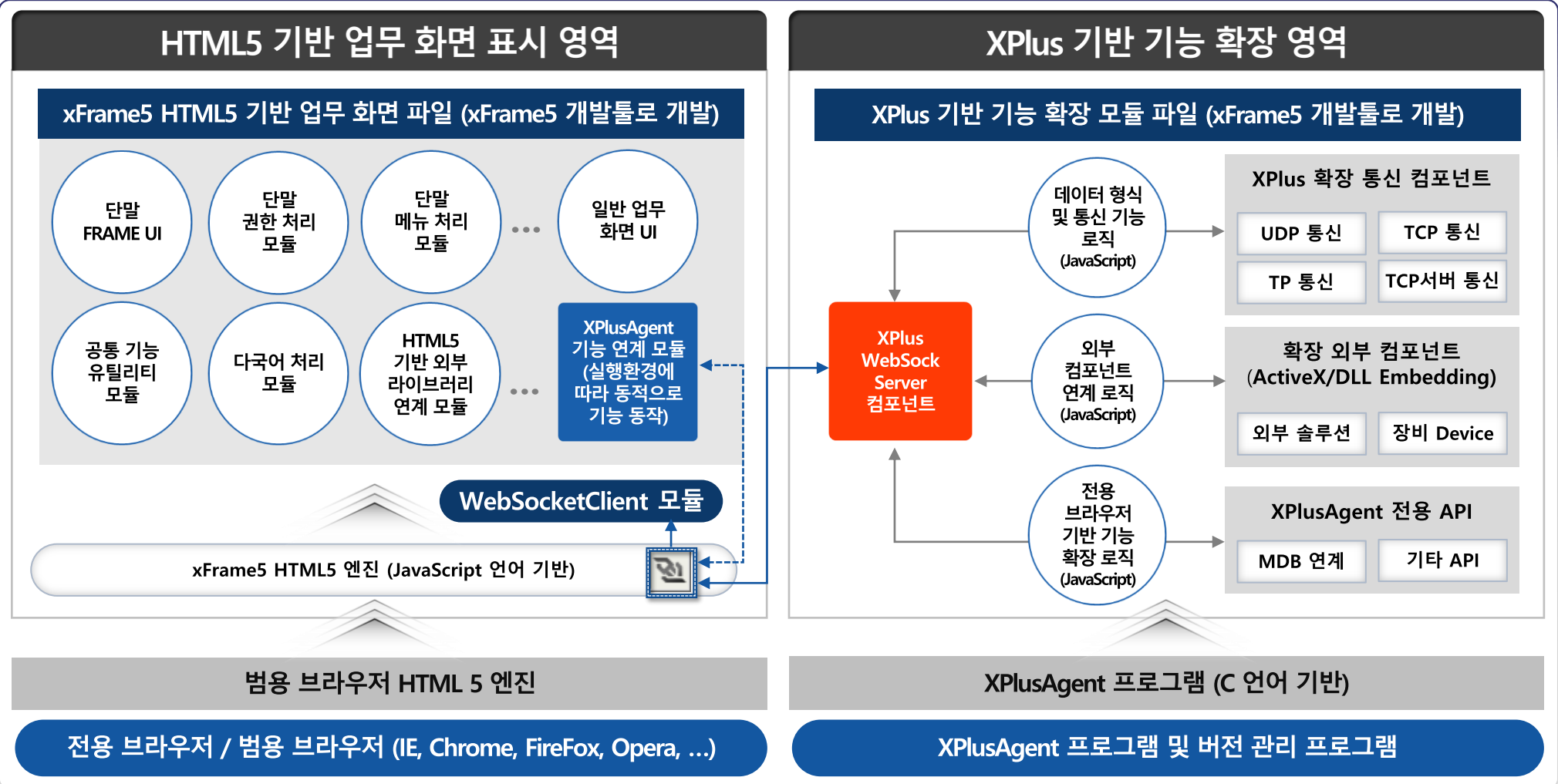


1. xFrame5 실행 환경 아키텍처

1.1. 실행 환경 아키텍처

▶ 아키텍처 상세 내용 - XPlus 확장 기능

xFrame5 솔루션의 범용 Browser 형태의 내부 구조는 아래와 같이 범용 브라우저 기반하에 HTML5 기반 업무 화면 표시 영역과 xPlus5 기반 기능 확장 영역으로 구성되며, 각 영역간 통신은 WebSocket 통신 프로토콜을 이용하여 수행된다.

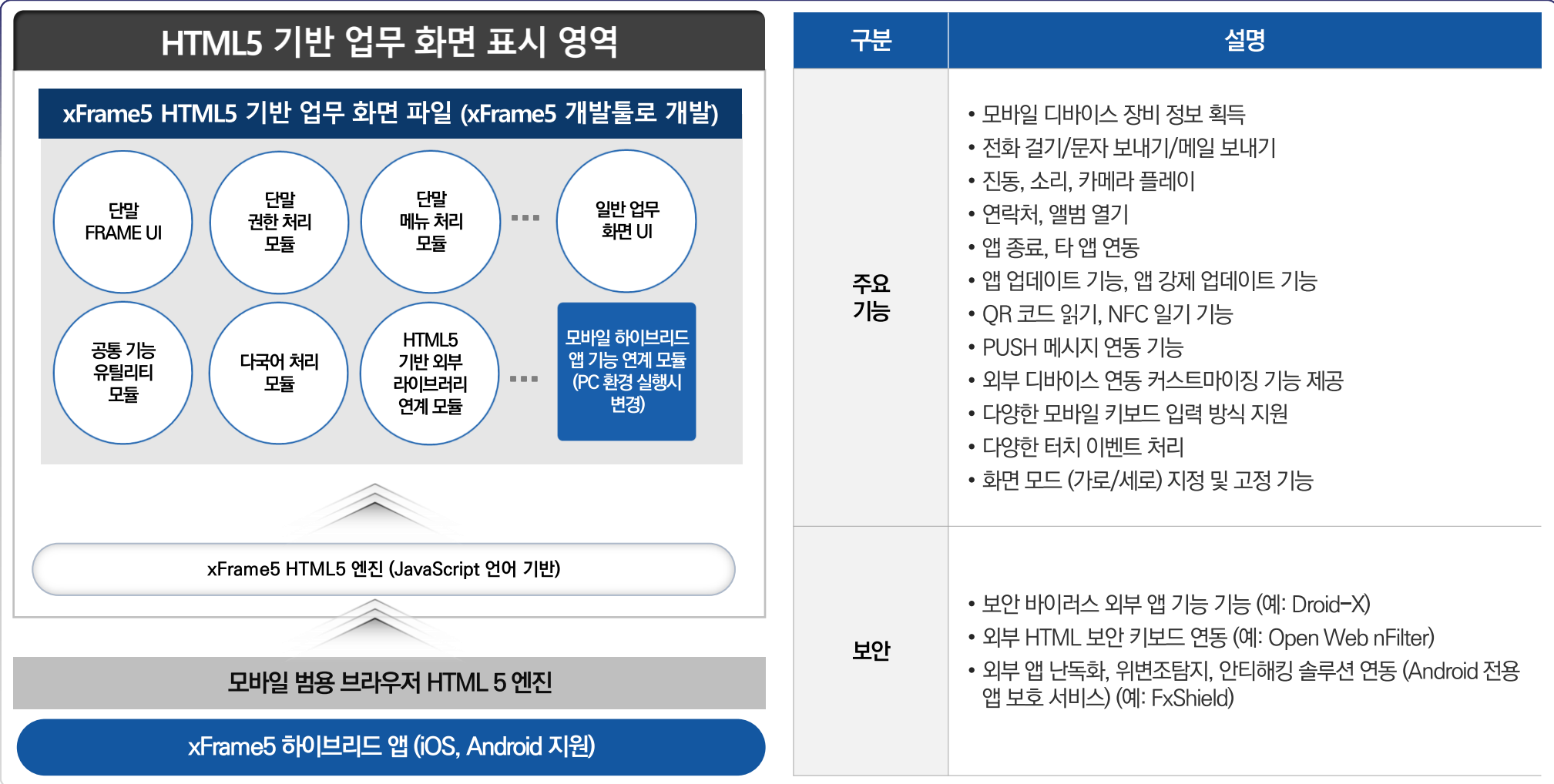


1. xFrame5 실행 환경 아키텍처

1.1. 실행 환경 아키텍처

▶ 아키텍처 상세 내용 - 모바일 하이브리드 앱

모바일 하이브리드 앱은 일반 모바일 WEB 환경에서 제공하지 못하는 다양한 네이티브(Native) 앱 기능을 제공하기 위해서, HTML5 환경에서 JavaScript로 연계를 위한 모바일 하이브리드 앱 기능 연계 모듈을 제공합니다.

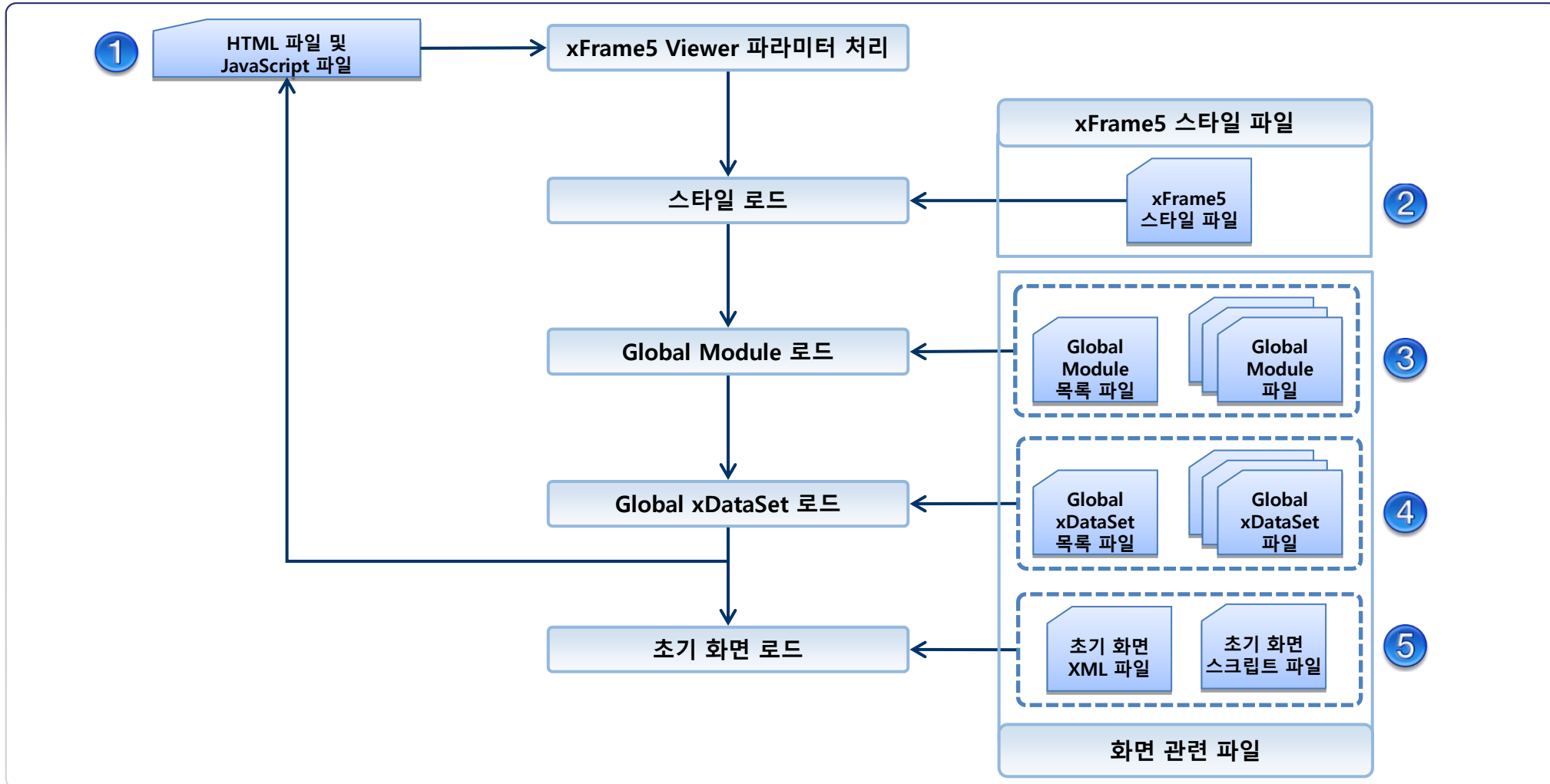


1. xFrame5 실행 환경 아키텍처

1.2. 초기 화면 실행 아키텍처

▶ 아키텍처 상세 내용

xFrame5 솔루션의 초기 화면 로딩 절차 및 실행 절차 아키텍처는 아래와 같다.



1. xFrame5 실행 환경 아키텍처

1.2. 초기 화면 실행 아키텍처

▶ 아키텍처 상세 설명

xFrame5 솔루션의 초기 화면 로딩 절차 및 실행 절차 아키텍처는 아래와 같다.

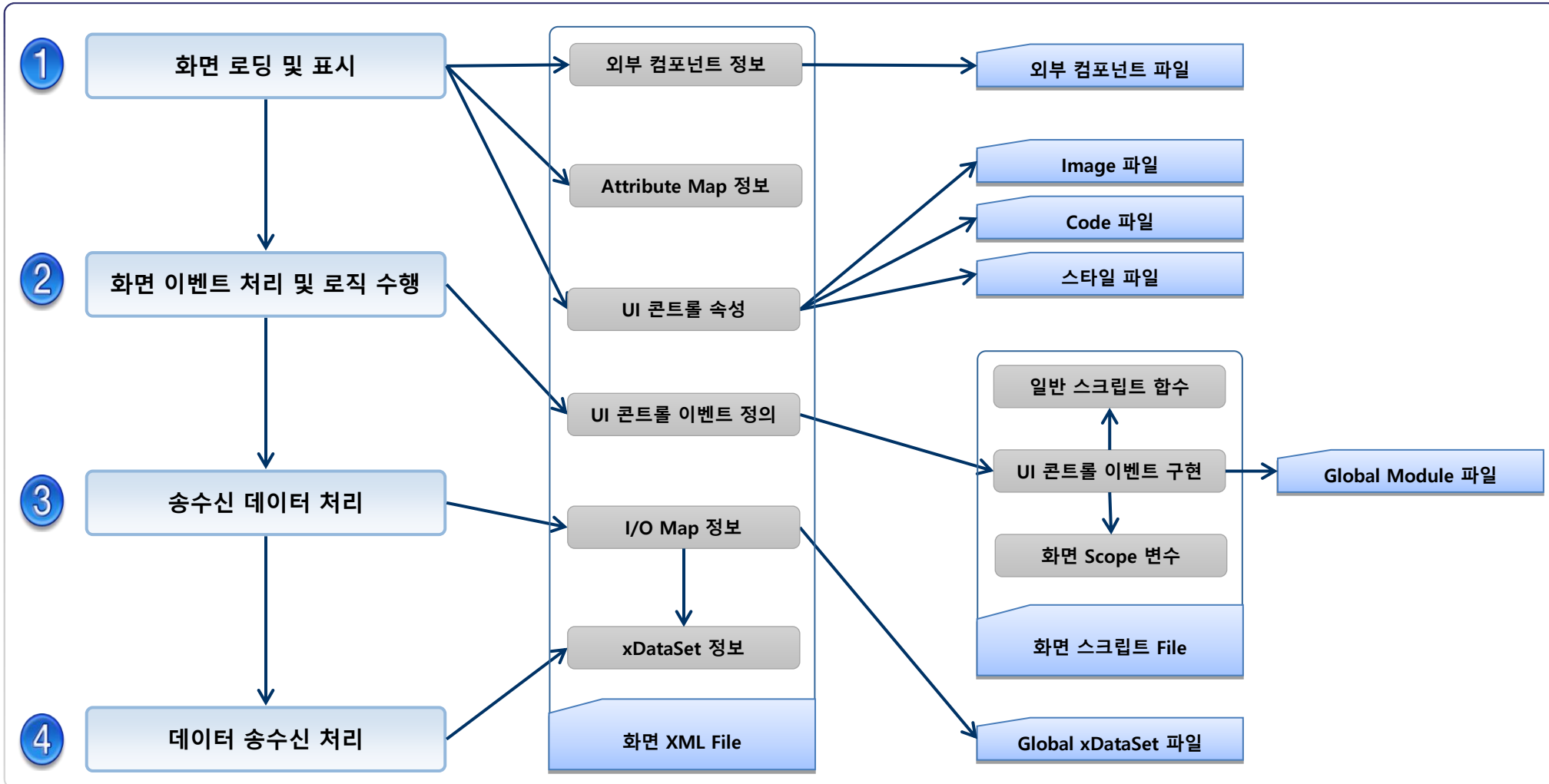
번호	구분	설명
1	xFrame Viewer 파라미터 처리	<ul style="list-style-type: none">▪ xFrameViewer를 품고 있는 HTML 또는 JavaScript 파일에서 xFrameViewer에게 전달되는 파라미터 정보를 처리한다.▪ 파라미터에는 화면 정보를 로드하는 방법 및 경로 정보가 포함되어 있다.
2	스타일 로드	<ul style="list-style-type: none">▪ xFrame5 프로젝트에서 정의한 스타일 파일을 로드한다.
3	Global Module 로드	<ul style="list-style-type: none">▪ 각각의 업무 화면에서 공통적으로 호출되어 사용되는 Global Module 정보를 Global Module 목록 파일 정보를 이용하여 로드한다.
4	Global xDataSet 로드	<ul style="list-style-type: none">▪ 각각의 업무 화면에서 공통적으로 사용할 수 있는 Global xDataSet 정보를 Global xDataSet 목록 파일 정보를 이용하여 로드한다.
5	초기 화면 로드	<ul style="list-style-type: none">▪ xFrameViewer의 파라미터 중에 "MAINFRAMESCREEN"에 해당하는 화면을 로드한다.

1. xFrame5 실행 환경 아키텍처

1.3. 화면 실행 아키텍처

▶ 아키텍처 상세 내용

xFrame5 솔루션의 업무 화면 로딩 절차 및 실행 절차 아키텍처는 아래와 같다.



1. xFrame5 실행 환경 아키텍처

1.3. 화면 실행 아키텍처

▶ 아키텍처 상세 설명

xFrame5 솔루션의 업무 화면 로딩 절차 및 실행 절차 아키텍처는 아래와 같다.

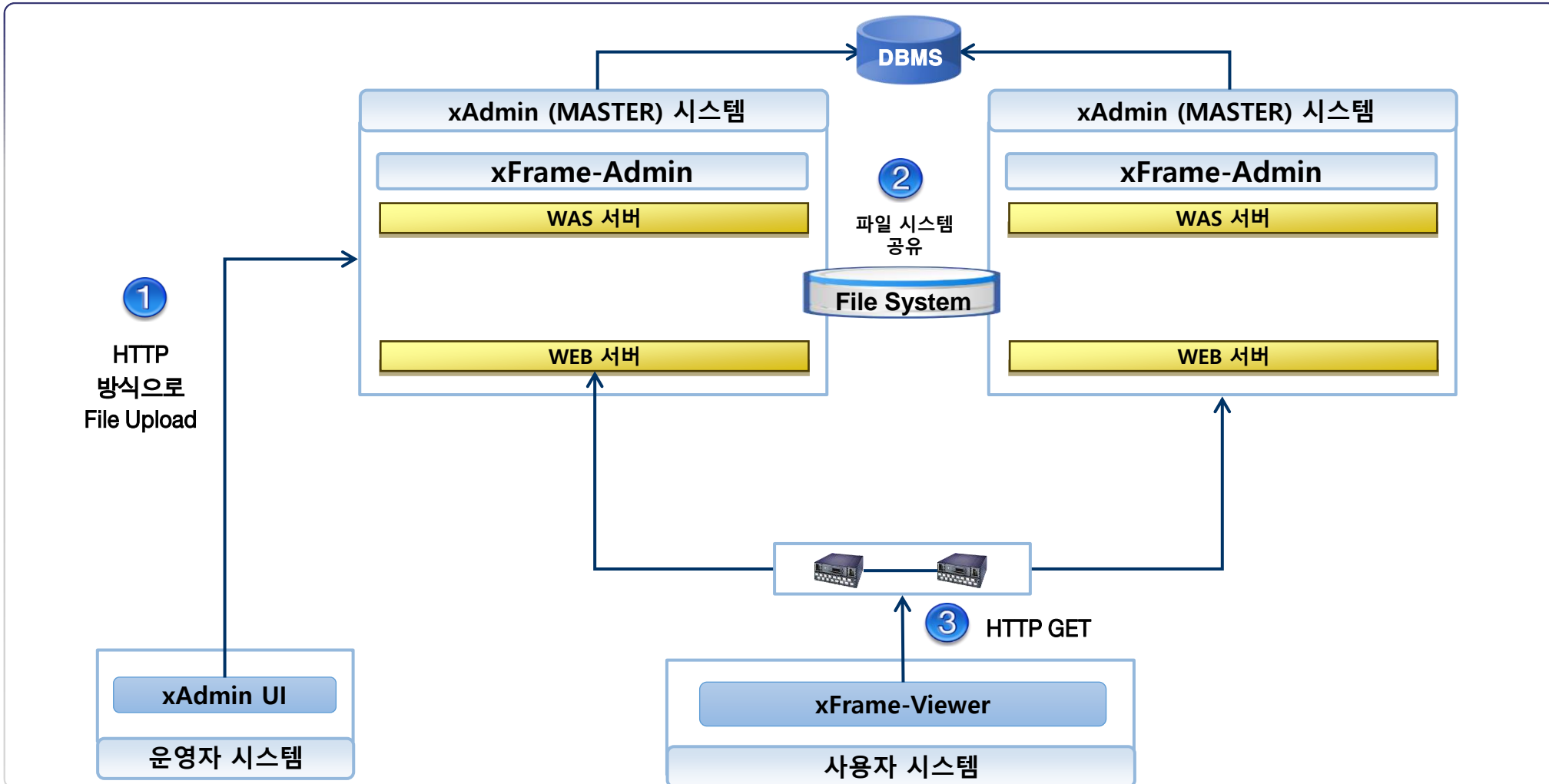
번호	구분	설명
1	화면 로딩 및 표시	<ul style="list-style-type: none">▪ 화면 로딩시 해당 화면 XML 파일내의 "외부 컴포넌트 정보", "Attribute Map 정보", "UI 컨트롤 속성" 정보를 분석하여 외부 리소스 파일들을 화면내에 로딩하고 표시한다.▪ "Attribute Map 정보"를 이용하여 UI 컨트롤의 초기 속성을 설정한다.
2	화면 이벤트 처리 및 로직 수행	<ul style="list-style-type: none">▪ 화면 XML 파일 내에 UI 컨트롤 이벤트 정의에 정의된 이벤트가 발생하면, 화면 스크립트 파일의 해당 함수가 호출된다.▪ 화면 스크립트 파일의 함수는 화면 내 변수 및 "Global Module"내에 있는 변수 및 함수를 이용하여 비즈니스 로직을 수행한다.
3	송수신 데이터 처리	<ul style="list-style-type: none">▪ 송수신 데이터 처리를 업무 서버로 데이터를 송수신하기 전에 송수신될 데이터의 정보를 이용해서 데이터를 조립/처리하는 과정이다.▪ "I/O Map 정보"에 정의된 데이터를 "xDataSet 정보"와 "Global xDataSet 파일" 정보를 이용해서 처리한다.
4	데이터 송수신 처리	<ul style="list-style-type: none">▪ 업무 서버와 xFrame-Viewer내에 내장된 통신 어댑터 또는 외부 통신 어댑터를 이용해서 데이터 송수신을 수행한다.

2. xAdmin5 배포환경 아키텍처

2.1. WEB/WAS가 하나의 시스템으로 구성 및 공유 디스크 사용

▶ 아키텍처 상세 내용

상세 아키텍처는 아래와 같다.



2. xAdmin5 배포환경 아키텍처

2.1. WEB/WAS가 하나의 시스템으로 구성 및 공유 디스크 사용

▶ 아키텍처 상세 설명

상세 아키텍처는 아래와 같다.

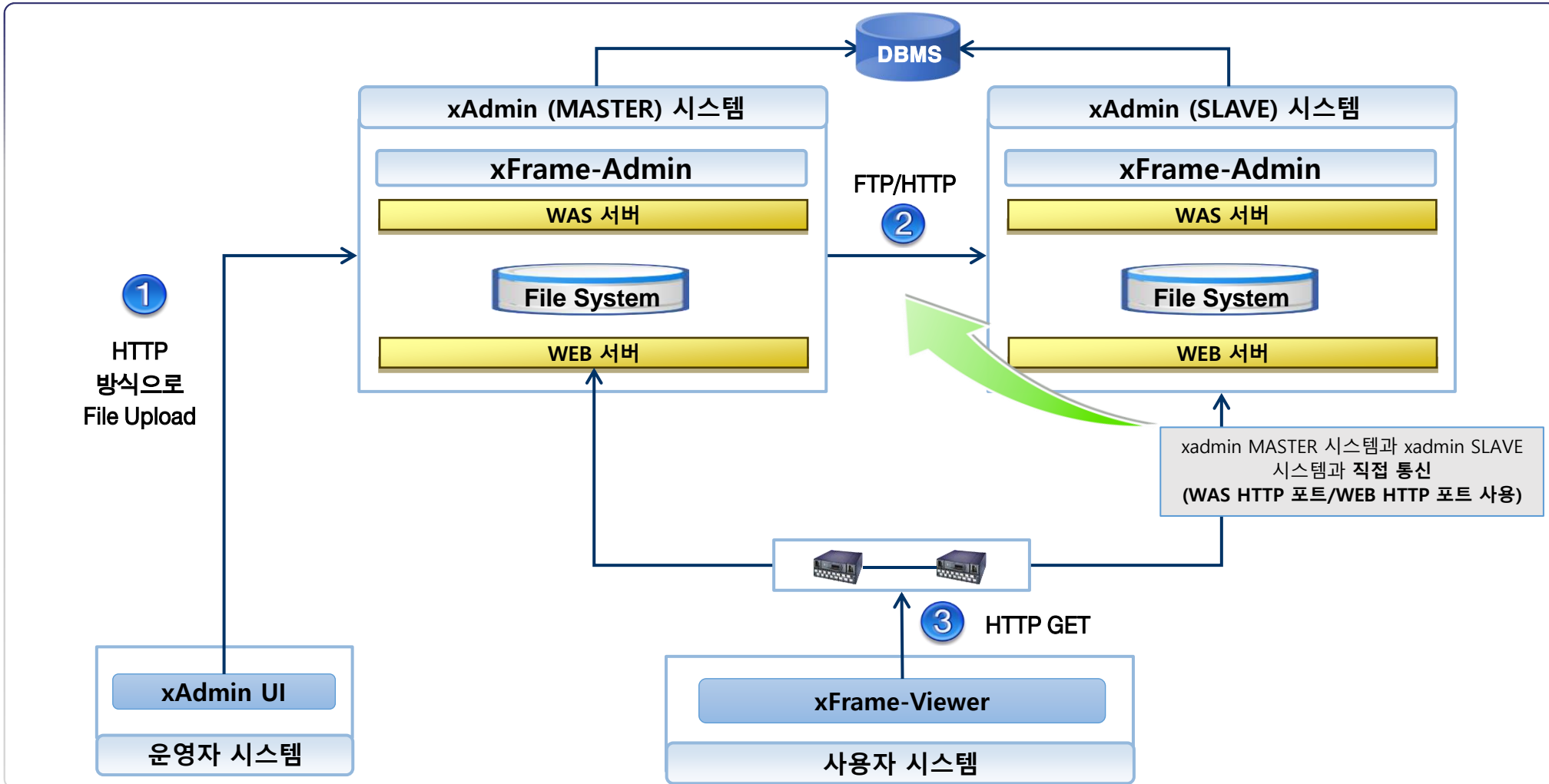
번호	구분	설명
1	파일 이행	<ul style="list-style-type: none">▪ XAdmin의 로컬 파일 업로드 또는 이관 업로드 기능을 통해 "운영자 시스템" 로컬 시스템의 파일을 업로드 하거나, 테스트 시스템의 파일을 업로드합니다.▪ XAdmin의 등록/결재/이행 기능을 통해 파일을 이행합니다.▪ 각 시스템은 XAdmin은 서버 관리 기능을 통해서 모두 "MASTER" 시스템으로 등록합니다.
2	파일 시스템 공유	<ul style="list-style-type: none">▪ 각 시스템은 외장 디스크 볼륨을 공유하여 사용합니다.
3	파일 다운로드 및 접근	<ul style="list-style-type: none">▪ L4 스위치의 가상 IP에 접속하여 WEB 서버를 통해서 필요한 파일들을 "사용자 시스템"에 다운로드합니다.

2. xAdmin5 배포환경 아키텍처

2.2. WEB/WAS가 하나의 시스템으로 구성 및 로컬 디스크 사용

▶ 아키텍처 상세 내용

상세 아키텍처는 아래와 같다.



2. xAdmin5 배포환경 아키텍처

2.2. WEB/WAS가 하나의 시스템으로 구성 및 로컬 디스크 사용

▶ 아키텍처 상세 설명

상세 아키텍처는 아래와 같다.

번호	구분	설명
1	파일 이행	<ul style="list-style-type: none"> ▪ 운영자는 XAdmin UI를 통해서 "MASTER" 시스템으로 등록된 시스템에 접속합니다. ▪ XAdmin의 로컬 파일 업로드 또는 이관 업로드 기능을 통해 "운영자 시스템" 로컬 시스템의 파일을 업로드 하거나, 테스트 시스템의 파일을 업로드합니다. ▪ XAdmin의 등록/결재/이행 기능을 통해 파일을 이행합니다. ▪ 각 시스템은 XAdmin은 서버 관리 기능을 통해서 모두 한 시스템을 "MASTER" 시스템으로 다른 시스템은 "SLAVE" 시스템으로 등록합니다. ▪ "MASTER" 시스템에만 백업본이 관리됩니다.
2	파일 시스템 동기화	<ul style="list-style-type: none"> ▪ 마스터 시스템에서 파일 이행시 "SLAVE" 시스템으로 등록된 시스템과 HTTP 통신 또는 FTP 통신을 통해 파일을 동기화 합니다. ▪ 동기화 방식은 XAdmin을 통해서 지정할 수 있습니다. ▪ FTP 통신 방식을 사용할 경우, FTP 서버가 설치되어 있어야 하며, XAdmin을 통해서 FTP 사용자 ID와 Password를 등록해야 합니다.
3	파일 다운로드 및 접근	<ul style="list-style-type: none"> ▪ L4 스위치의 가상 IP에 접속하여 WEB 서버를 통해서 필요한 파일들을 "사용자 시스템"에 다운로드합니다.

2. xAdmin5 배포환경 아키텍처

2.2. WEB/WAS가 하나의 시스템으로 구성 및 로컬 디스크 사용

▶ 장애 처리 절차

상세 아키텍처는 아래와 같다.

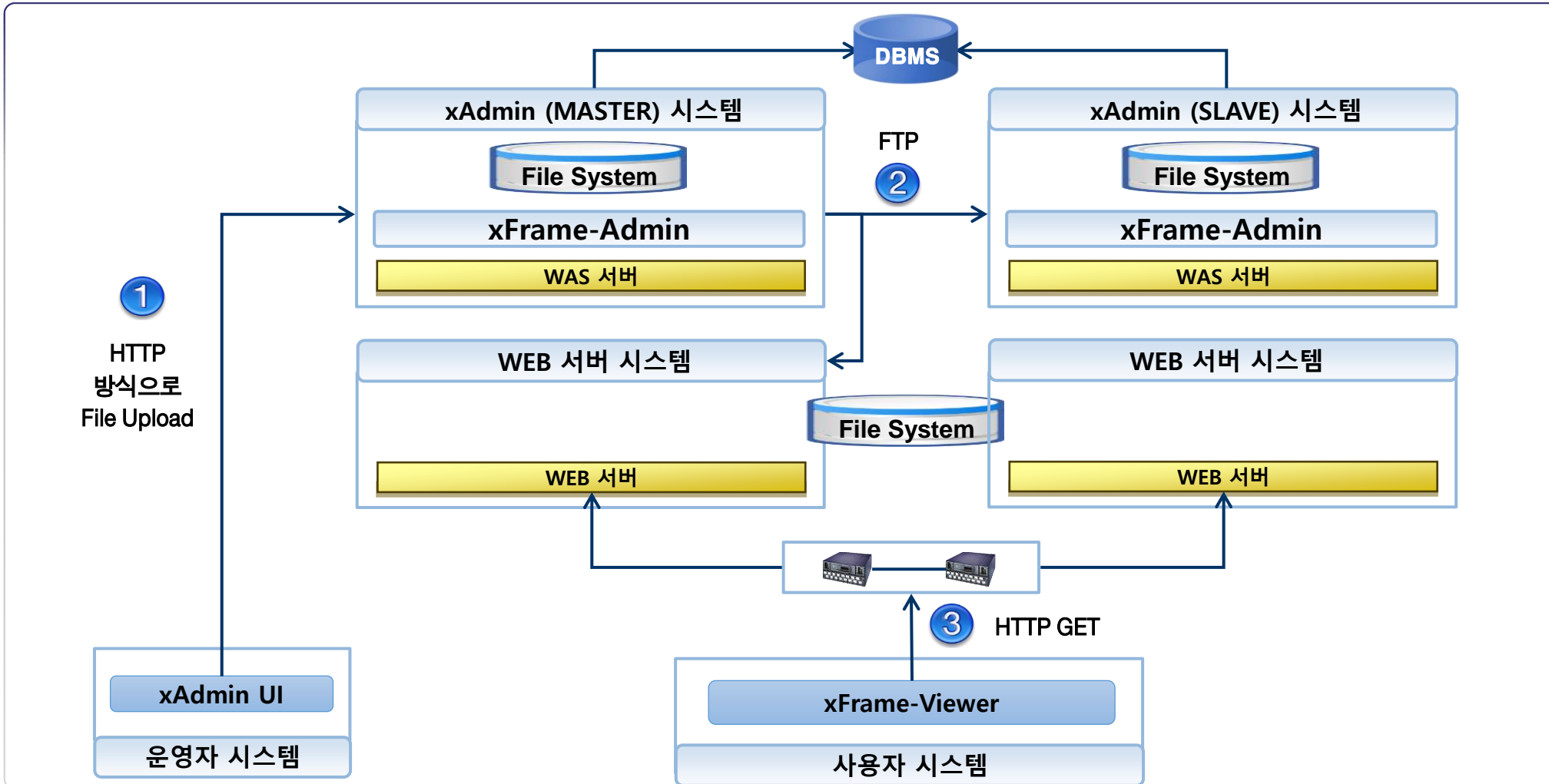
번호	구분	설명
1	마스터 서버가 장애인 경우	<ul style="list-style-type: none">▪ "SLAVE" 시스템에 XAdmin으로 접속하여, "SLAVE" 시스템의 정보를 "MASTER" 시스템으로 변경합니다.▪ XAdmin의 로컬 파일 업로드 또는 이관 업로드 기능을 통해 "운영자 시스템" 로컬 시스템의 파일을 업로드 하거나, 테스트 시스템의 파일을 업로드합니다.▪ XAdmin의 등록/결재/이행 기능을 통해 파일을 이행합니다.
2	장애 발생 시스템이 복구된 경우	<ul style="list-style-type: none">▪ "MASTER"로 설정된 시스템에 접속하여, 이행 이력을 확인하여, 장애 발생 시간동안 배포된 파일 목록을 확인합니다.▪ 마스터 관리 메뉴를 통해서 해당 파일들을 로컬로 다운로드 합니다.▪ 로컬로 다운로드된 파일들을 다시 업로드/등록/결재/이행 절차를 수행하여 배포합니다.
3	서버 구성 정보 복구	<ul style="list-style-type: none">▪ "MASTER" 시스템의 정보를 변경한 경우, 다시 서버 정보를 원복합니다.

2. xAdmin5 배포환경 아키텍처

2.3. WEB/WAS가 분리된 시스템으로 구성 및 공유 디스크 사용

▶ 아키텍처 상세 내용

상세 아키텍처는 아래와 같다.



2. xAdmin5 배포환경 아키텍처

2.3. WEB/WAS가 분리된 시스템으로 구성

▶ 아키텍처 상세 설명

상세 아키텍처는 아래와 같다.

번호	구분	설명
1	파일 이행	<ul style="list-style-type: none">▪ 운영자는 XAdmin UI를 통해서 "MASTER" 시스템으로 등록된 시스템에 접속합니다.▪ XAdmin의 로컬 파일 업로드 또는 이관 업로드 기능을 통해 "운영자 시스템" 로컬 시스템의 파일을 업로드 하거나, 테스트 시스템의 파일을 업로드합니다.▪ XAdmin의 등록/결재/이행 기능을 통해 파일을 이행합니다.▪ 각 시스템은 XAdmin은 서버 관리 기능을 통해서 모두 한 시스템을 "MASTER" 시스템으로 다른 시스템은 "SLAVE" 시스템으로 등록합니다.▪ "MASTER" 시스템에만 백업본이 관리됩니다.
2	파일 시스템 동기화	<ul style="list-style-type: none">▪ 마스터 시스템에서 파일 이행시 "SLAVE" 시스템으로 등록된 시스템, WEB 서버 시스템으로 설정된 서버와 FTP 통신을 통해 파일을 동기화 합니다.▪ HTTP 동기화 방식은 지원하지 않습니다.▪ FTP 통신 방식을 사용할 경우, FTP 서버가 설치되어 있어야 하며, XAdmin을 통해서 FTP 사용자 ID와 Password를 등록해야 합니다.
3	파일 다운로드 및 접근	<ul style="list-style-type: none">▪ L4 스위치의 가상 IP에 접속하여 WEB 서버를 통해서 필요한 파일들을 "사용자 시스템"에 다운로드합니다.

2. xAdmin5 배포환경 아키텍처

2.3. WEB/WAS가 분리된 시스템으로 구성

▶ 장애 처리 절차

상세 아키텍처는 아래와 같다.

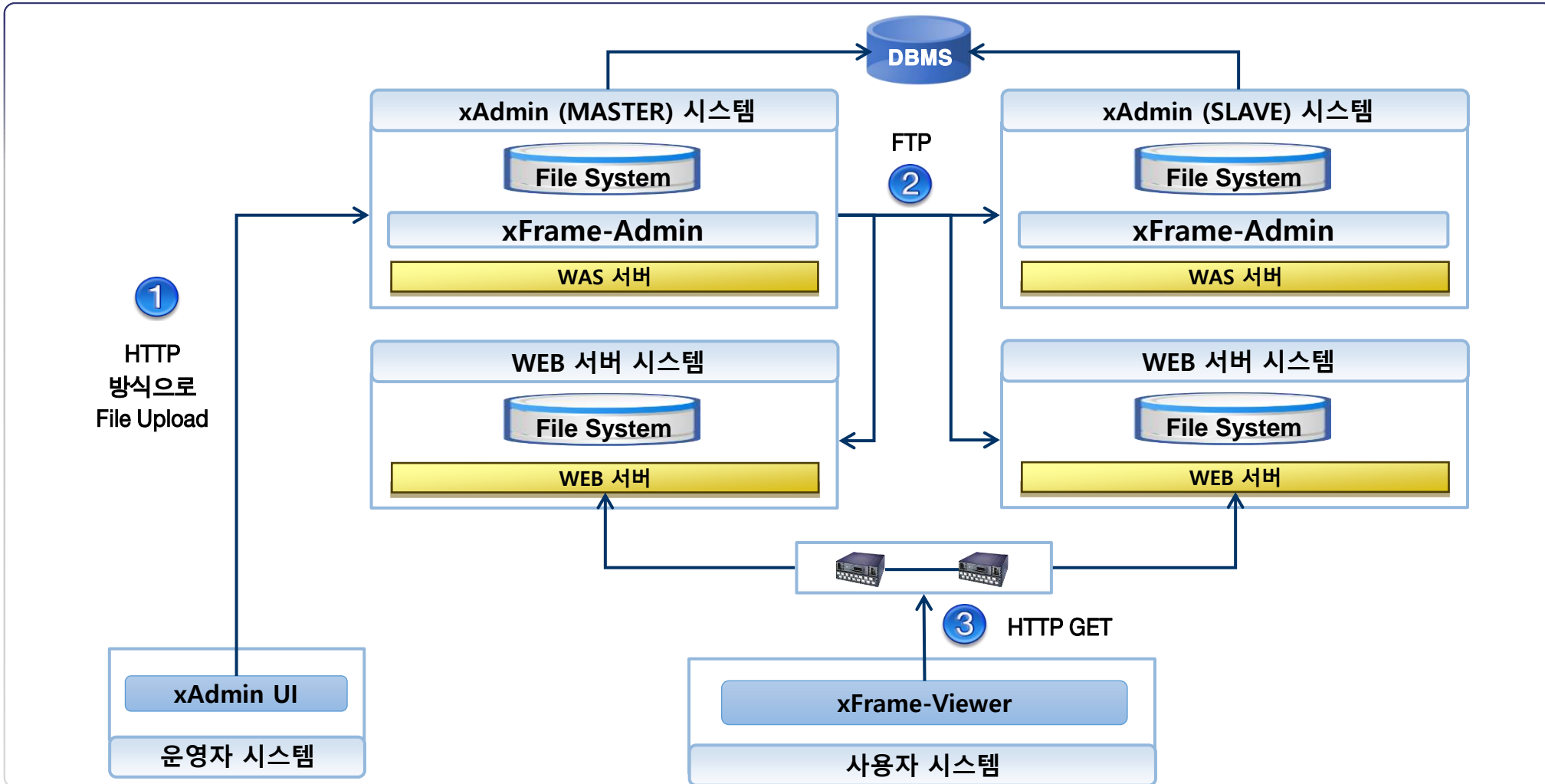
번호	구분	설명
1	마스터 서버가 장애인 경우	<ul style="list-style-type: none">▪ "SLAVE" 시스템에 XAdmin으로 접속하여, "SLAVE" 시스템의 정보를 "MASTER" 시스템으로 변경합니다.▪ XAdmin의 로컬 파일 업로드 또는 이관 업로드 기능을 통해 "운영자 시스템" 로컬 시스템의 파일을 업로드 하거나, 테스트 시스템의 파일을 업로드합니다.▪ XAdmin의 등록/결재/이행 기능을 통해 파일을 이행합니다.
2	장애 발생 시스템이 복구된 경우	<ul style="list-style-type: none">▪ "MASTER"로 설정된 시스템에 접속하여, 이행 이력을 확인하여, 장애 발생 시간동안 배포된 파일 목록을 확인합니다.▪ 마스터 관리 메뉴를 통해서 해당 파일들을 로컬로 다운로드 합니다.▪ 로컬로 다운로드된 파일들을 다시 업로드/등록/결재/이행 절차를 수행하여 배포합니다.
3	서버 구성 정보 복구	<ul style="list-style-type: none">▪ "MASTER" 시스템의 정보를 변경한 경우, 다시 서버 정보를 원복합니다.

2. xAdmin5 배포환경 아키텍처

2.3. WEB/WAS가 분리된 시스템으로 구성 및 로컬 디스크 사용

▶ 아키텍처 상세 내용

상세 아키텍처는 아래와 같다.



2. xAdmin5 배포환경 아키텍처

2.3. WEB/WAS가 분리된 시스템으로 구성된 경우

▶ 아키텍처 상세 설명

상세 아키텍처는 아래와 같다.

번호	구분	설명
1	파일 이행	<ul style="list-style-type: none">▪ 운영자는 XAdmin UI를 통해서 "MASTER" 시스템으로 등록된 시스템에 접속합니다.▪ XAdmin의 로컬 파일 업로드 또는 이관 업로드 기능을 통해 "운영자 시스템" 로컬 시스템의 파일을 업로드 하거나, 테스트 시스템의 파일을 업로드합니다.▪ XAdmin의 등록/결재/이행 기능을 통해 파일을 이행합니다.▪ 각 시스템은 XAdmin은 서버 관리 기능을 통해서 모두 한 시스템을 "MASTER" 시스템으로 다른 시스템은 "SLAVE" 시스템으로 등록합니다.▪ "MASTER" 시스템에만 백업본이 관리됩니다.
2	파일 시스템 동기화	<ul style="list-style-type: none">▪ 마스터 시스템에서 파일 이행시 "SLAVE" 시스템으로 등록된 시스템, WEB 서버 시스템으로 설정된 서버와 FTP 통신을 통해 파일을 동기화 합니다.▪ HTTP 동기화 방식은 지원하지 않습니다.▪ FTP 통신 방식을 사용할 경우, FTP 서버가 설치되어 있어야 하며, XAdmin을 통해서 FTP 사용자 ID와 Password를 등록해야 합니다.
3	파일 다운로드 및 접근	<ul style="list-style-type: none">▪ L4 스위치의 가상 IP에 접속하여 WEB 서버를 통해서 필요한 파일들을 "사용자 시스템"에 다운로드합니다.

2. xAdmin5 배포환경 아키텍처

2.3. WEB/WAS가 분리된 시스템으로 구성된 경우

▶ 장애 처리 절차

상세 아키텍처는 아래와 같다.

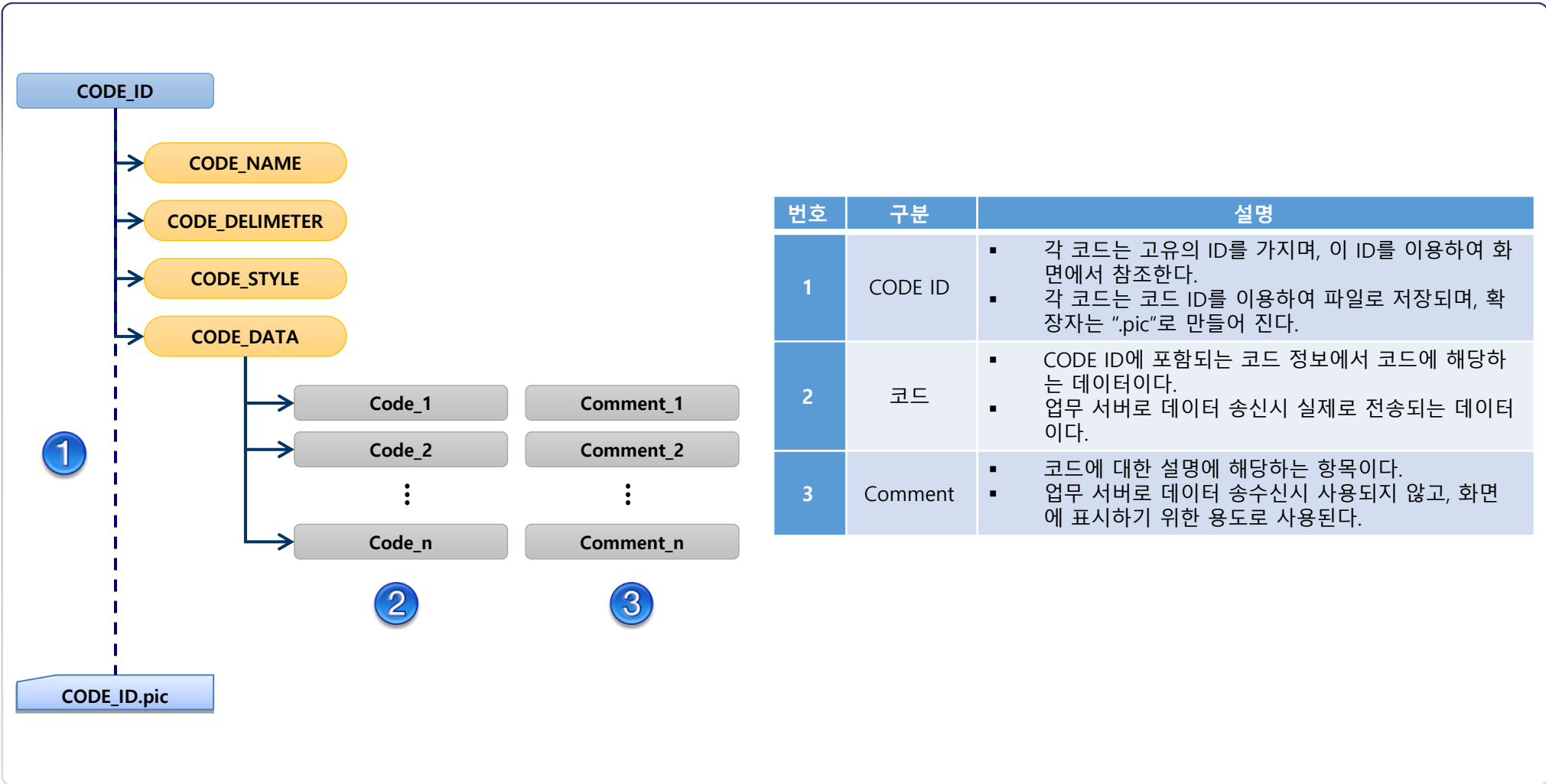
번호	구분	설명
1	마스터 서버가 장애인 경우	<ul style="list-style-type: none">▪ "SLAVE" 시스템에 XAdmin으로 접속하여, "SLAVE" 시스템의 정보를 "MASTER" 시스템으로 변경합니다.▪ XAdmin의 로컬 파일 업로드 또는 이관 업로드 기능을 통해 "운영자 시스템" 로컬 시스템의 파일을 업로드 하거나, 테스트 시스템의 파일을 업로드합니다.▪ XAdmin의 등록/결재/이행 기능을 통해 파일을 이행합니다.
2	장애 발생 시스템이 복구된 경우	<ul style="list-style-type: none">▪ "MASTER"로 설정된 시스템에 접속하여, 이행 이력을 확인하여, 장애 발생 시간동안 배포된 파일 목록을 확인합니다.▪ 마스터 관리 메뉴를 통해서 해당 파일들을 로컬로 다운로드 합니다.▪ 로컬로 다운로드된 파일들을 다시 업로드/등록/결재/이행 절차를 수행하여 배포합니다.
3	서버 구성 정보 복구	<ul style="list-style-type: none">▪ "MASTER" 시스템의 정보를 변경한 경우, 다시 서버 정보를 원복합니다.

3. xFrame5 컴포넌트 아키텍처

3.1. 코드 정보 아키텍처

▶ 아키텍처 상세 내용

상세 아키텍처는 아래와 같다.



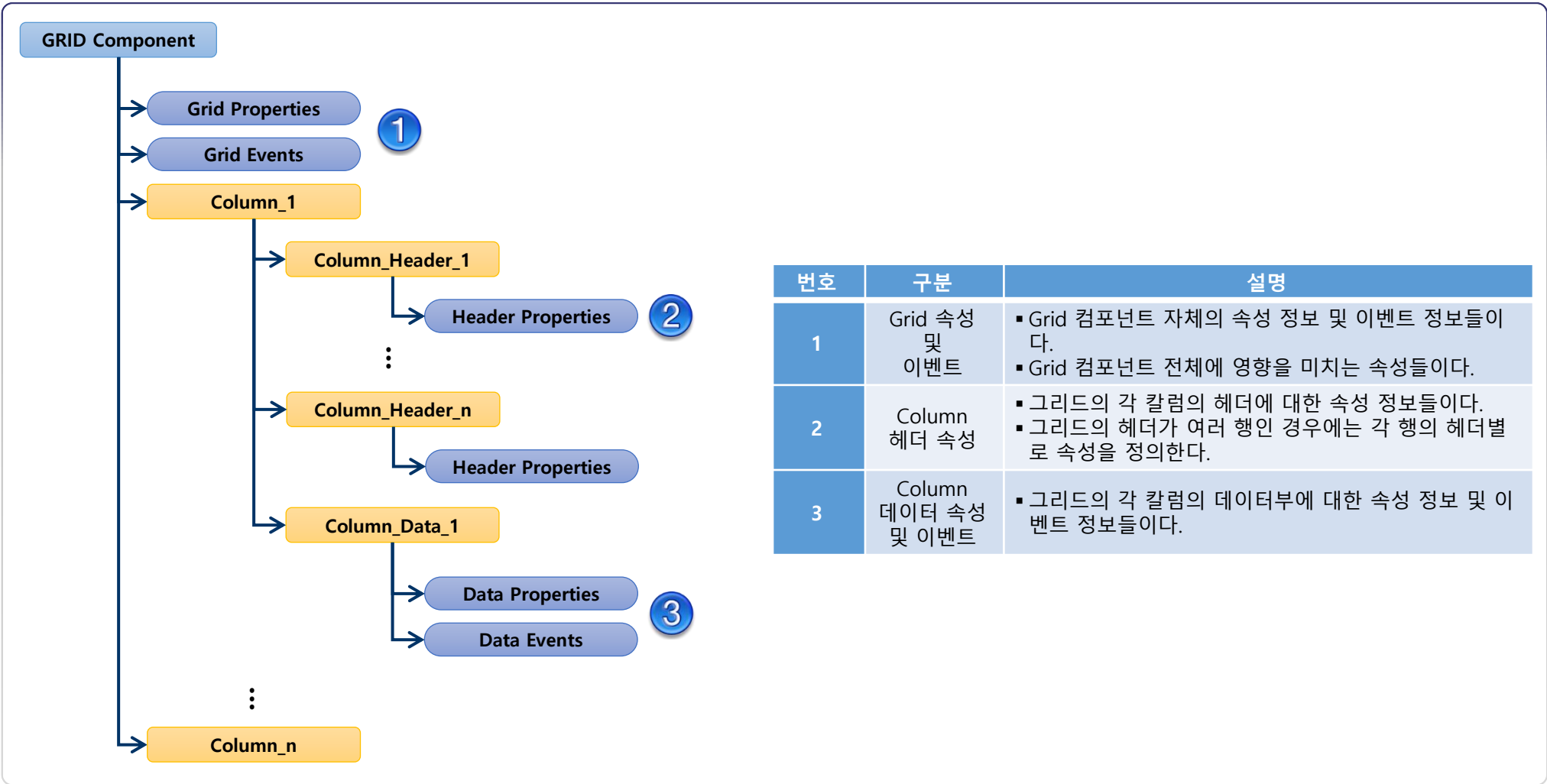
번호	구분	설명
1	CODE ID	<ul style="list-style-type: none"> 각 코드는 고유의 ID를 가지며, 이 ID를 이용하여 화면에서 참조한다. 각 코드는 코드 ID를 이용하여 파일로 저장되며, 확장자는 ".pic"로 만들어 진다.
2	코드	<ul style="list-style-type: none"> CODE ID에 포함되는 코드 정보에서 코드에 해당하는 데이터이다. 업무 서버로 데이터 송신시 실제로 전송되는 데이터이다.
3	Comment	<ul style="list-style-type: none"> 코드에 대한 설명에 해당하는 항목이다. 업무 서버로 데이터 송수신시 사용되지 않고, 화면에 표시하기 위한 용도로 사용된다.

3. xFrame5 컴포넌트 아키텍처

3.2. 그리드 정보 아키텍처

▶ 아키텍처 상세 내용

상세 아키텍처는 아래와 같다.



번호	구분	설명
1	Grid 속성 및 이벤트	<ul style="list-style-type: none"> Grid 컴포넌트 자체의 속성 정보 및 이벤트 정보들이다. Grid 컴포넌트 전체에 영향을 미치는 속성들이다.
2	Column 헤더 속성	<ul style="list-style-type: none"> 그리드의 각 칼럼의 헤더에 대한 속성 정보들이다. 그리드의 헤더가 여러 행인 경우에는 각 행의 헤더별로 속성을 정의한다.
3	Column 데이터 속성 및 이벤트	<ul style="list-style-type: none"> 그리드의 각 칼럼의 데이터부에 대한 속성 정보 및 이벤트 정보들이다.

3. xFrame5 컴포넌트 아키텍처

3.2. 그리드 정보 아키텍처

▶ 아키텍처 상세 내용

상세 아키텍처는 아래와 같다.

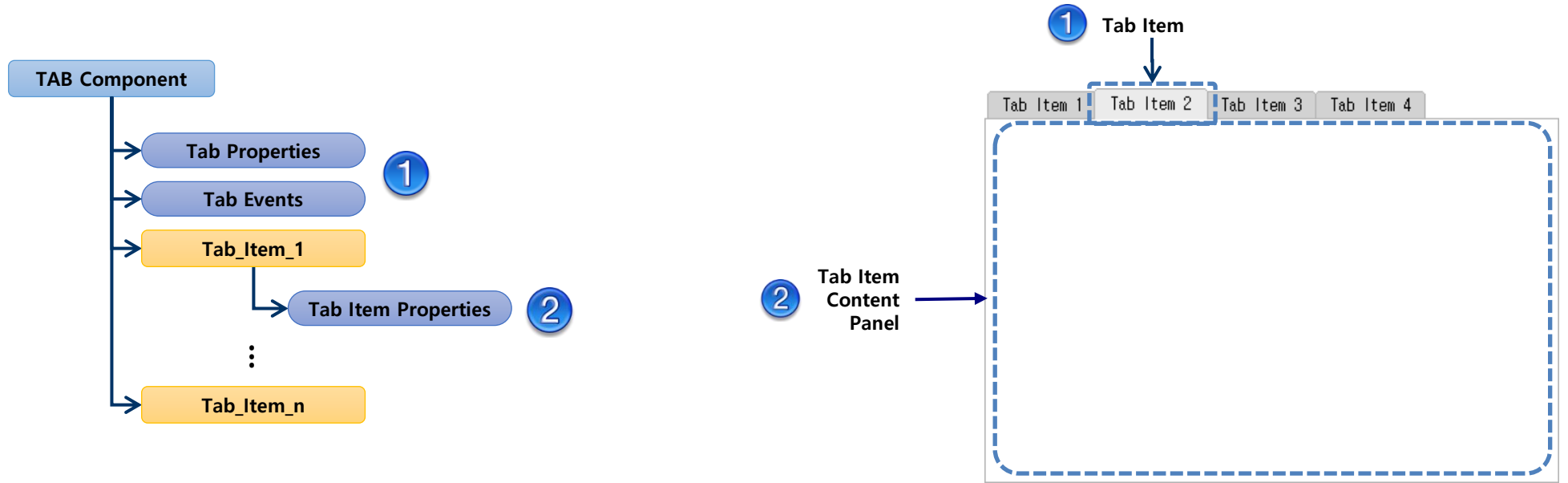
번호	구분	설명
1	Grid Header	▪ 그리드의 각 컬럼의 헤더 정보가 표시 되는 영역이다.
2	Grid Row	▪ 그리드의 한 행을 의미한다.
3	Grid Column	▪ 그리드의 각 열을 의미한다.
4	Grid Item	<ul style="list-style-type: none"> ▪ 그리드의 특정 행의 특정 컬럼을 의미한다. ▪ 그리드에서 제공하는 API를 이용하면 각 그리드 아이템 단위로 동작을 수행할 수 있다.

3. xFrame5 컴포넌트 아키텍처

3.3. 탭 정보 아키텍처

▶ 아키텍처 상세 내용

상세 아키텍처는 아래와 같다.



번호	구분	설명
1	Tab 속성 및 이벤트	<ul style="list-style-type: none"> Tab 컴포넌트 자체의 속성 정보 및 이벤트 정보들이다. Tab 컴포넌트 전체에 영향을 미치는 속성들이다.
2	Tab Item 속성	<ul style="list-style-type: none"> 탭의 각 아이템에 대한 속성 정보들이다. 탭의 아이템이 여러 개인 경우에는 각 탭 아이템별로 속성을 정의한다.

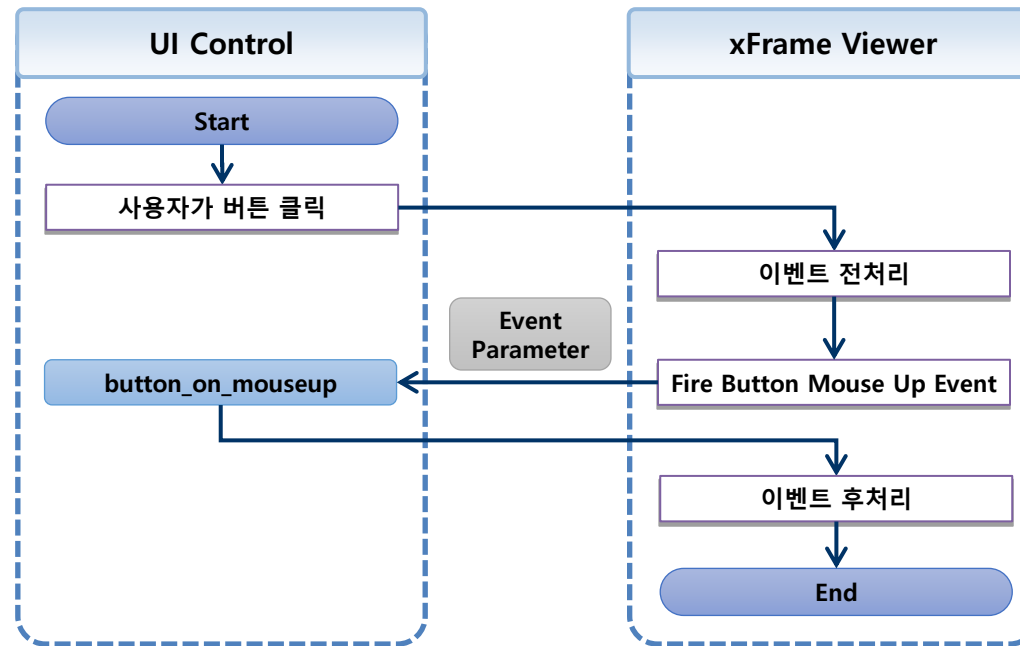
번호	구분	설명
1	Tab Item	<ul style="list-style-type: none"> Tab 컨트롤은 각각의 여러 개의 탭 아이템으로 구성된다.
2	Tab Content Panel	<ul style="list-style-type: none"> Tab 컨트롤의 각각의 탭 아이템은 아이템별로 콘텐츠 영역을 가진다. 탭 아이템 콘텐츠 영역에서는 직접 컴포넌트가 배치될 수도 있고, 다른 화면을 링크를 걸 수 있다.

4. xFrame5 이벤트 아키텍처

4.1. 단순 이벤트 아키텍처

▶ 아키텍처 상세 내용

상세 아키텍처는 아래와 같다.

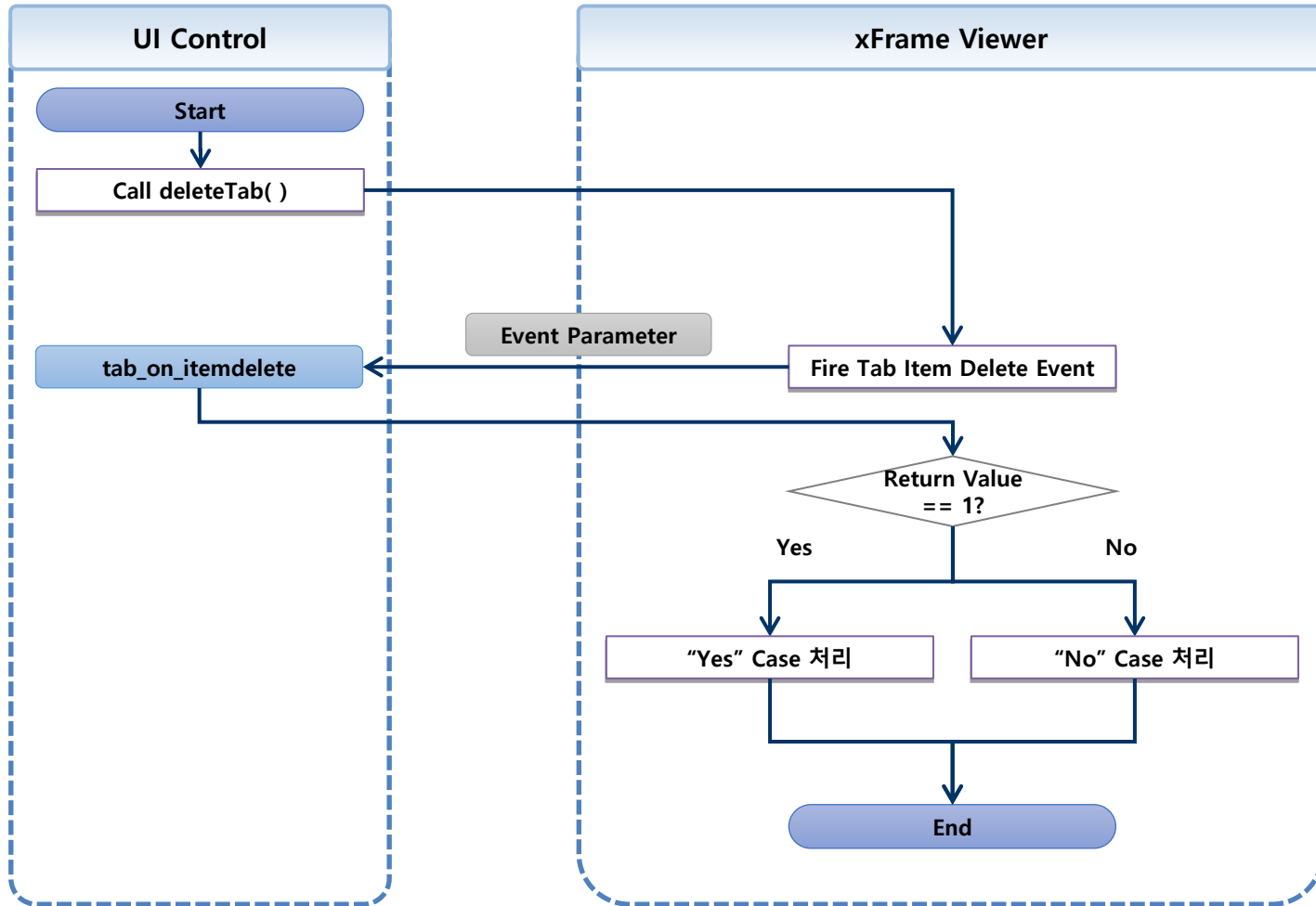


4. xFrame5 이벤트 아키텍처

4.2. 리턴 이벤트 아키텍처

▶ 아키텍처 상세 내용

상세 아키텍처는 아래와 같다.

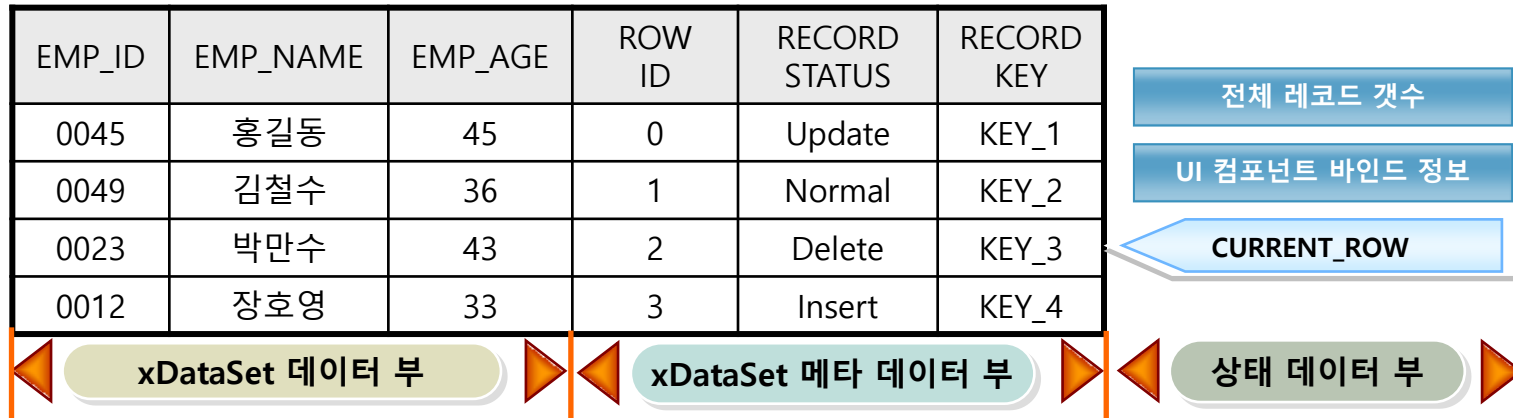


5. xFrame5 데이터셋 아키텍처

5.1. 데이터셋 내부 아키텍처

▶ 아키텍처 상세 내용

상세 아키텍처는 아래와 같다.

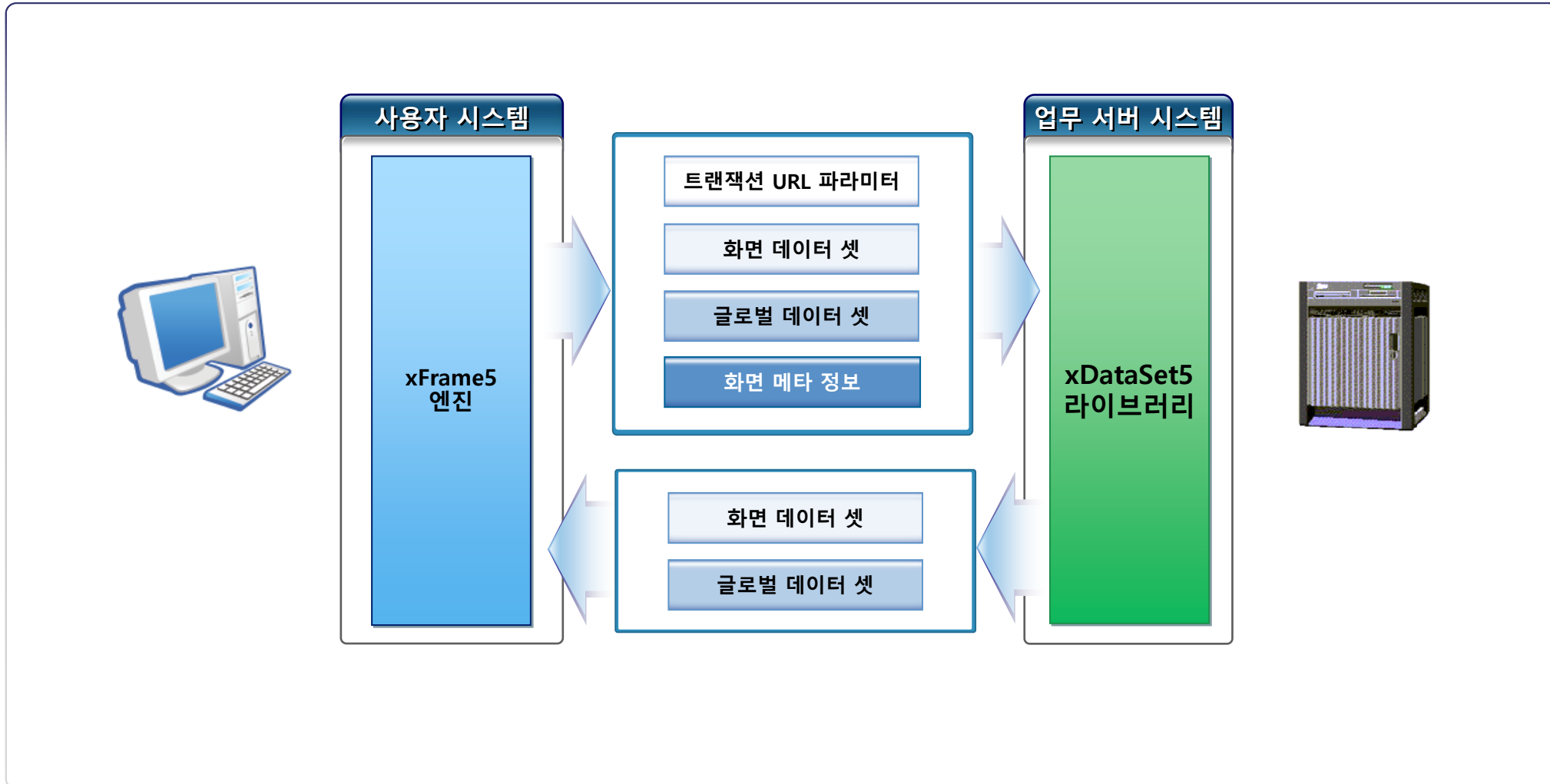


5. xFrame5 데이터셋 아키텍처

5.2. 데이터셋 통신 아키텍처

▶ 아키텍처 상세 내용

상세 아키텍처는 아래와 같다.

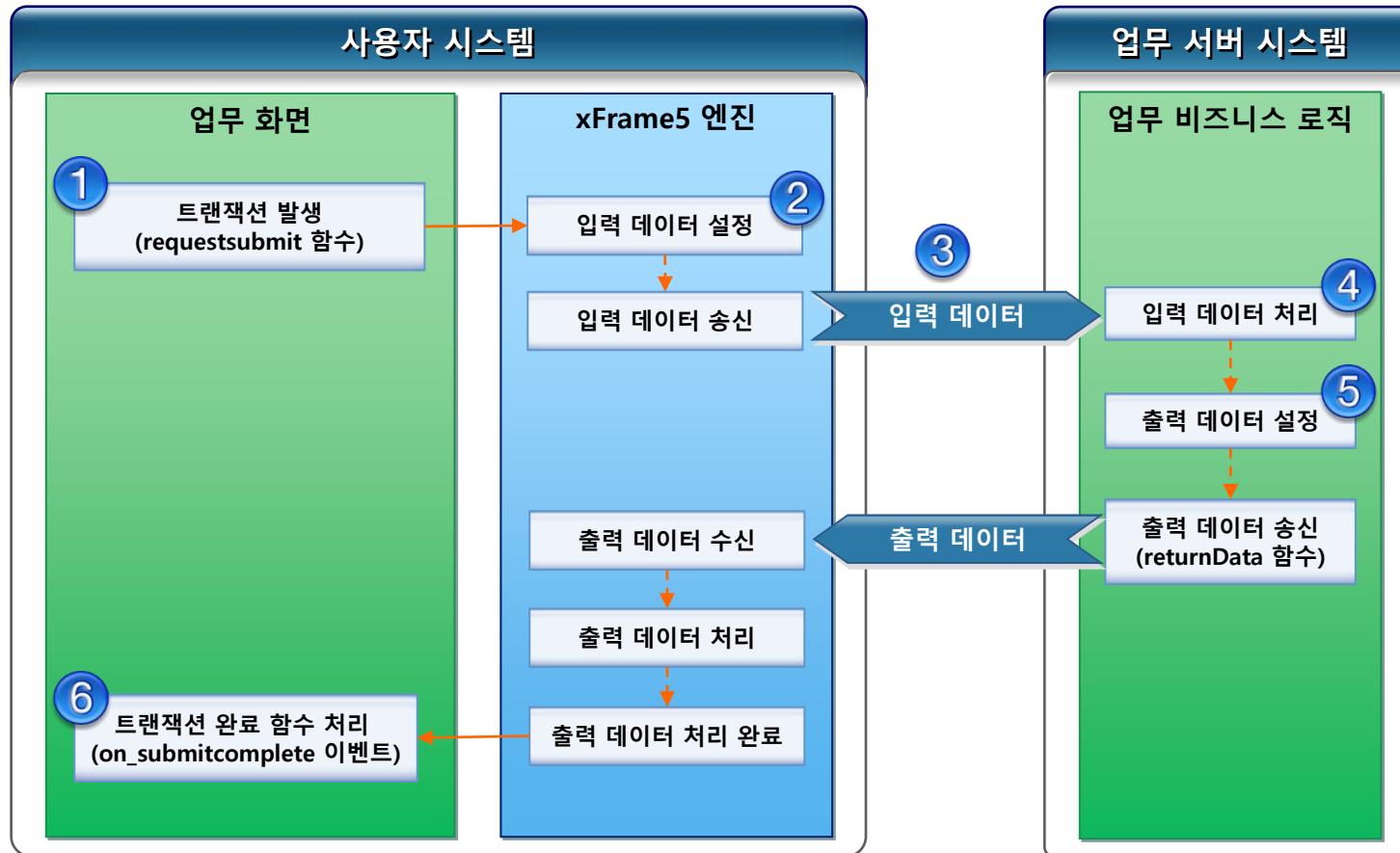


5. xFrame5 데이터셋 아키텍처

5.3. 데이터셋 흐름 아키텍처

▶ 아키텍처 상세 내용

상세 아키텍처는 아래와 같다.



감사합니다.

